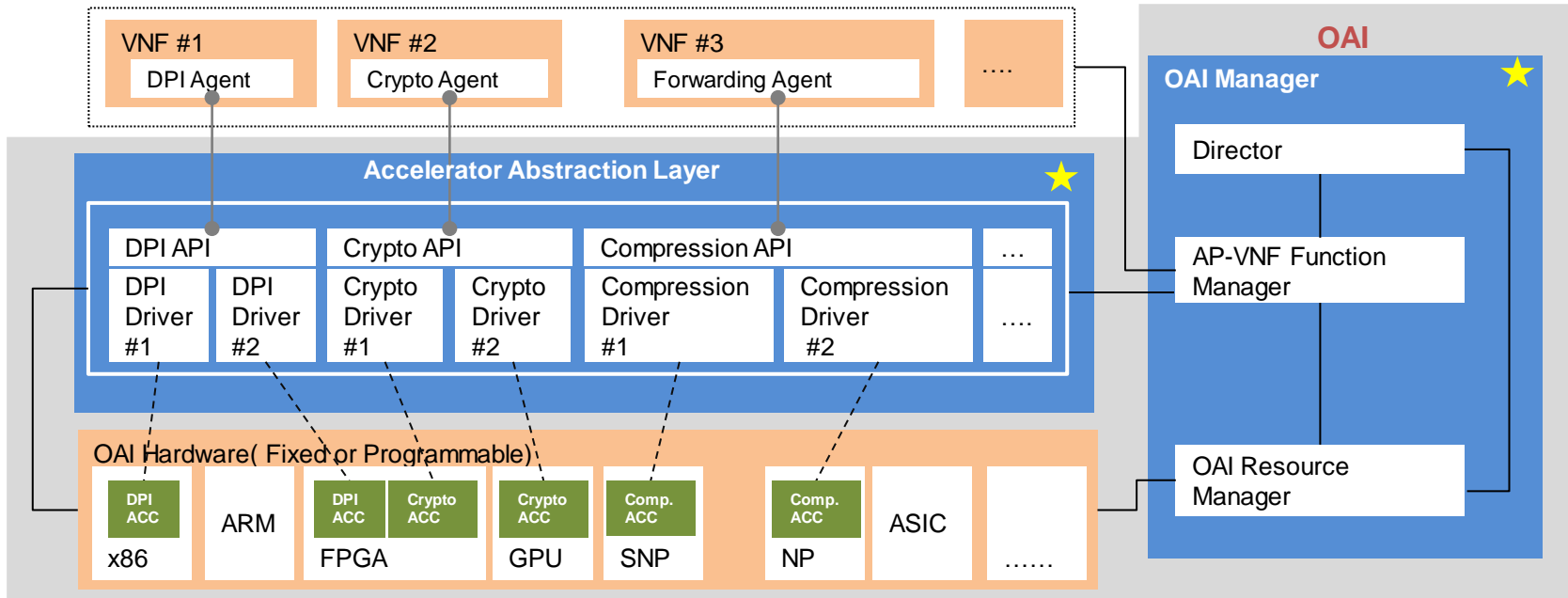


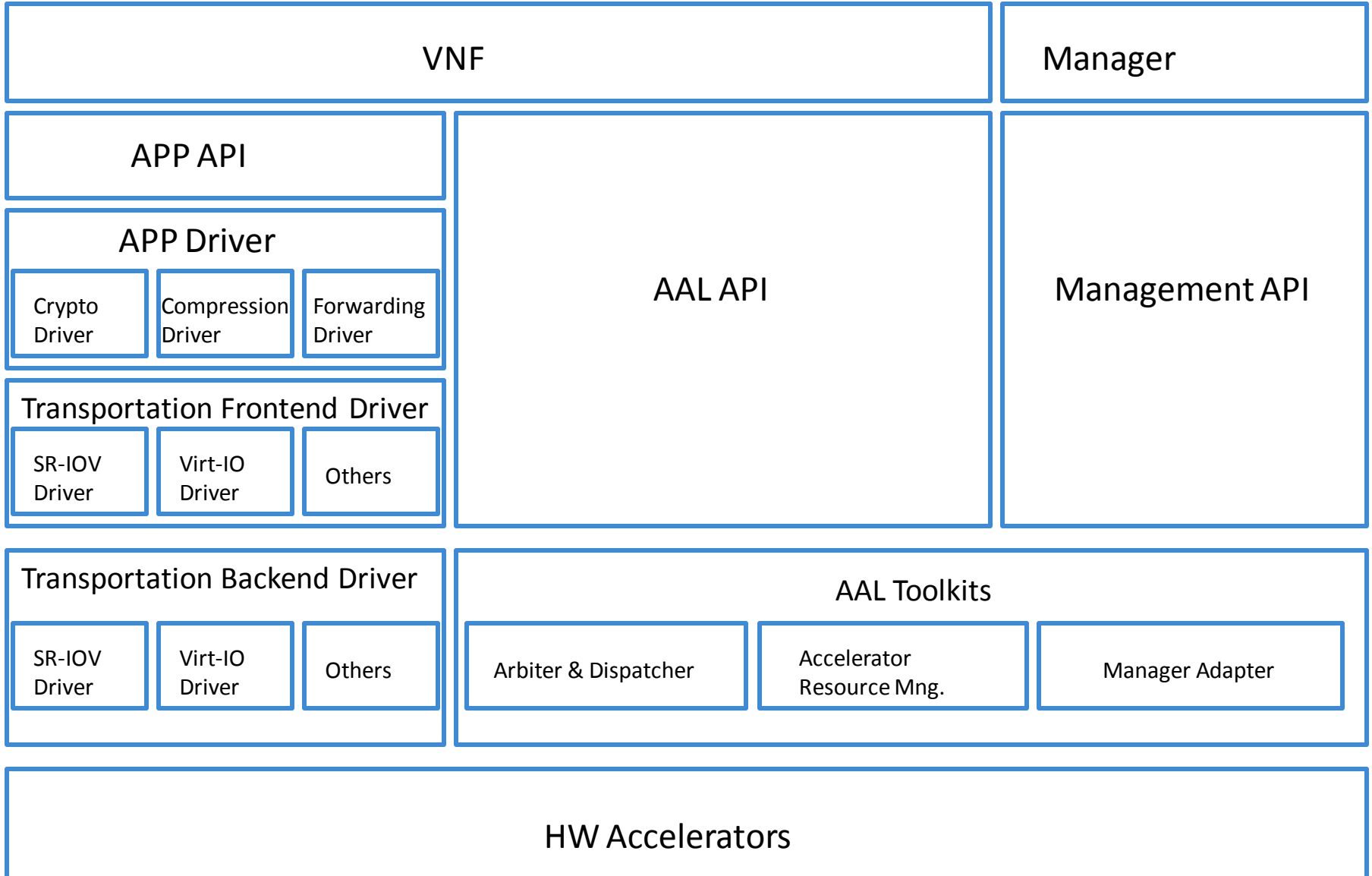
Uniform API Design for DPACC

Recap



- AAL : a set of open and common APIs, which shields the difference of accelerators as well as offer a uniform interface for VNFs. VNFs can involve these APIs to implement various of accelerated functions (e.g. Crypto, Compression, DPI and Forwarding)
- Manager : in charge of the management and the orchestration of accelerator resources and bridging between VNFs and accelerators.

Framework



Work Flows of VNF Acceleration

Phase I: Setting Up

- aal_find_device : Ask AAL whether or not there is a accelerator is available
input: Function type, such as Crypto, Compression... and other requirement
output: OK or No, potential accelerator ID if OK(the accelerator will not be created until the next API)
- aal_acquire_device: Ask AAL to enable the accelerator
Input: accelerator ID which is gotten before
Output: OK, which means the accelerator is working
No, the accelerator can not work

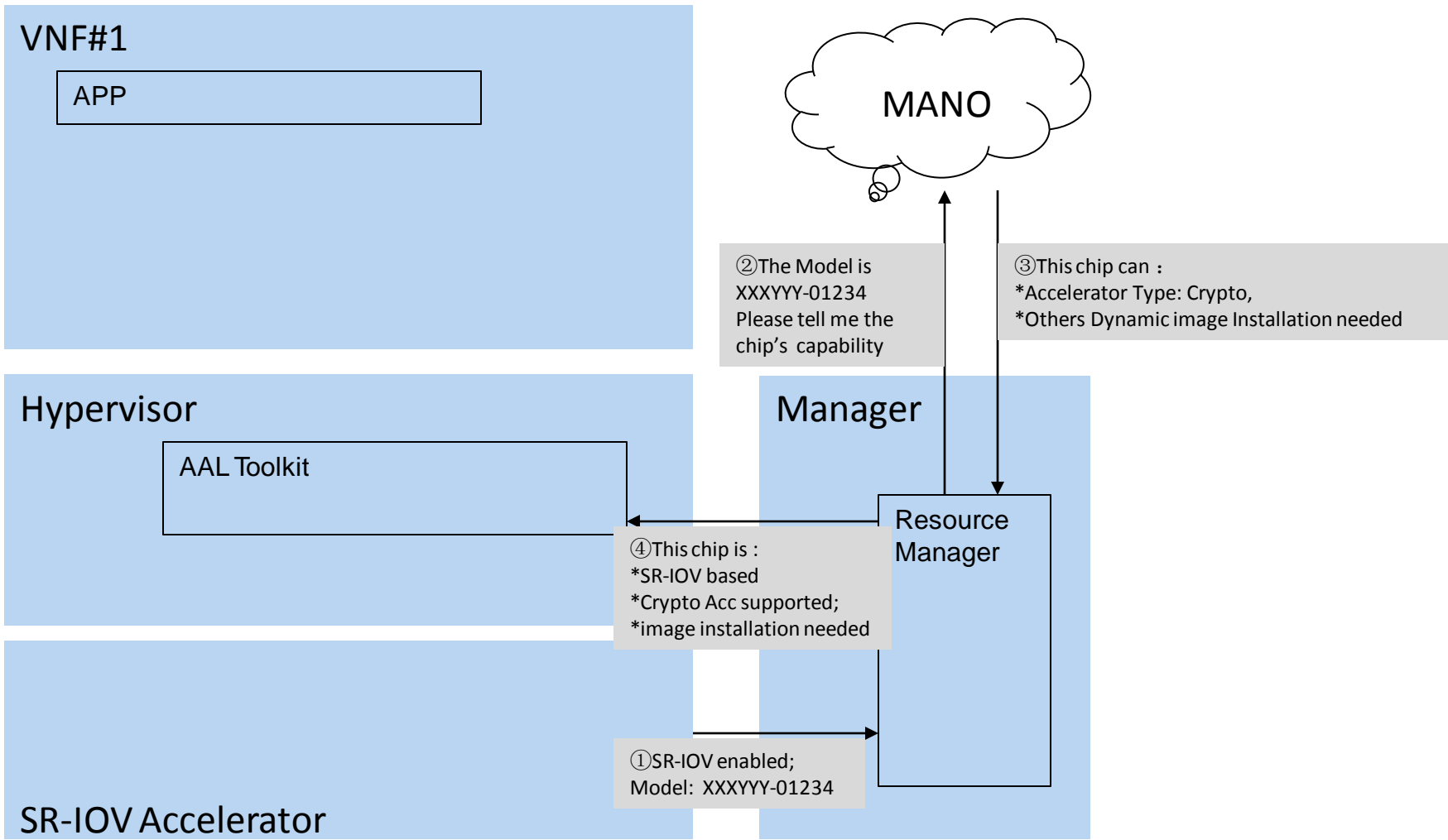
Phase II: Accelerating

- aal_crypto_handle_create: create a process handle with the accelerator
input: accelerator ID which is bound with this VNF
output: OK or No, handle ID
- aal_encode_crypto: ask for a crypto process (example)
Input: handle ID, clear text
Output: encrypted text
- aal_handle_release: release a handle
Input: handle ID
Output: OK or NO

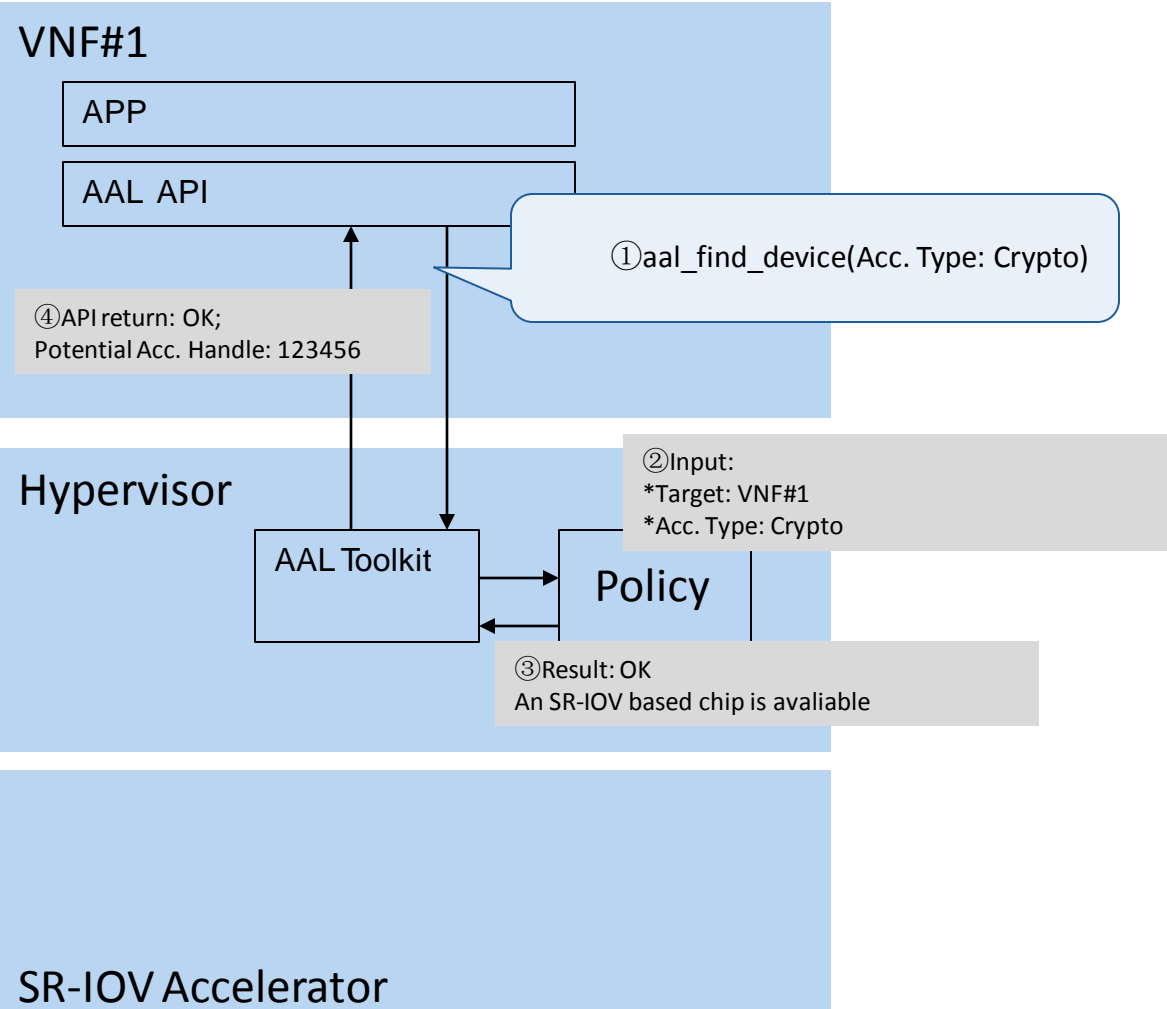
Phase III: Release Accelerator

- aal_release: unbind the accelerator
input: device ID
output: OK or No

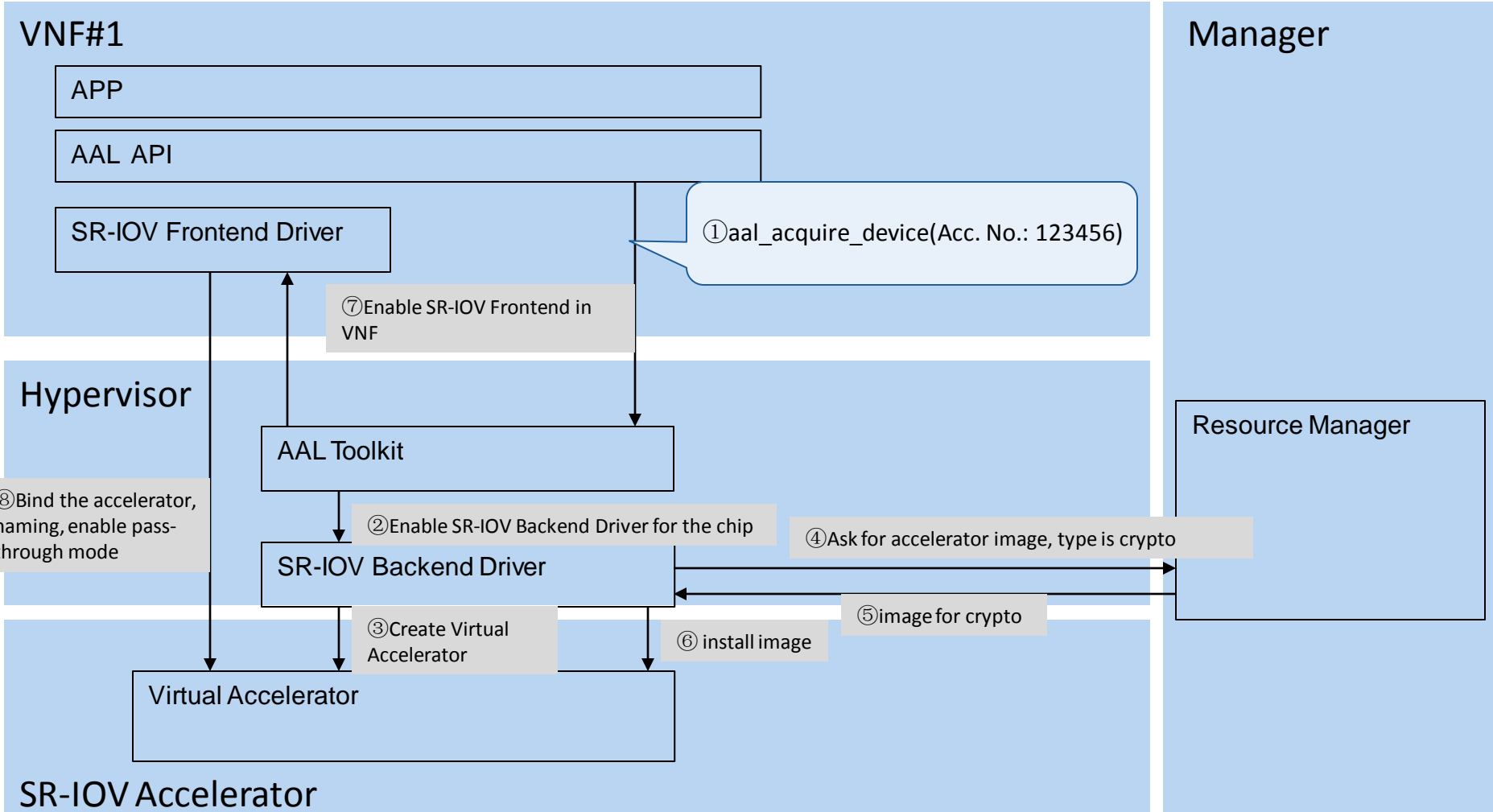
Initialization : Accelerator Registering (SR-IOV for instance)



Setting Up: Accelerator Finding (SR-IOV for instance)



Setting Up : Accelerator Acquiring (SR-IOV for instance)



Thank you!